Multimedia Systems
Lecture – 5

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

Digital Representation of Text

- Any type of data needs to be converted to its digital form (in the form 1s and 0s) to be represented in the Computer.

- Text (letters, symbols, and numerals) also needs to be encoded in digital form is what allows computers to manipulate and communicate text.

- Two character encoding standards define how characters are decoded from ones and zeros into the text you see on the screen right now, and into the different languages viewed every day on the World Wide Web.

- These two encoding standards are ASCII and Unicode.

ASCII

- The **American Standard Code for Information Interchange** (ASCII) was developed to create an international standard for encoding the Latin alphabet.
- In 1963, ASCII was adopted so information could be interpreted between computers; representing lower and upper letters, numbers, symbols, and some commands.
- Because ASCII is encoded using ones and zeros, the base 2 number system, it uses seven bits.
- Seven bits allow 2 to the power of 7 = 128 possible combinations of digits to encode a character.

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

How encoding ASCII works

- You already know how to convert between decimal and binary numbers
- You now need to turn letters into binary numbers
- Every character has a corresponding decimal number (for example, A → 65)
- ASCII uses 7 bits
- We use the first 7 columns of the conversion table to create 128 different numbers (from 0 to 127)

- Example: 1000001 gives us the number 65 (64+1), which corresponds to the letter A

64	32	16	8	4	2	1
1	0	0	0	0	0	1

- Here's how 'HELLO' is encoded in ASCII in binary:

Latin character	ASCII
H	1001000
E	1000101
L	1001100
L	1001100
O	1001111

Let's apply this theory in practice:

- Open Notepad, or whichever plain text editor you prefer
- Type a message and save it, e.g. 'data is beautiful'
- Look at the size of the file — mine is 18 bytes
- Now, add another word, e.g. 'data is SO beautiful'
- If you look at the file size again, you'll see that it has changed — my file is now 3 bytes larger (SO[SPACE]: the 'S', the 'O', and the space)

Extended ASCII

- Because ASCII encodes characters in 7 bits, moving to 8-bit computing technology meant there was one extra bit to be used.
- With this extra digit, **Extended ASCII** encoded up to 256 characters.
- However, the problem that developed was that countries that used different languages did different things with this extra capacity for encoding.
- Many countries added their own additional characters, and different numbers represented different characters in different languages.
- The problem of incompatible encoding systems became more urgent with the invention of the World Wide Web, as people shared digital documents all over the world, using multiple languages.
- To address the issue, the **Unicode Consortium** established a universal encoding system called **Unicode**.

Unicode

- Unicode (Unique, Universal, and Uniform character enCoding) is the new standard for representing characters of all the languages of the World.
- The latest version of Unicode contains a repertoire of more than 120,000 characters covering 129 modern and historic scripts, as well as multiple symbol sets.
- ASCII character encoding is a subset of Unicode.
- Unicode can be implemented by different character encodings. The Unicode standard defines UTF-8, UTF-16 and UTF-32.
- So, these use between 8 and 32 bits per character and has the advantage that it represents many more unique characters than ASCII because of the larger number of bits available to store a character code.
- It uses the same codes as ASCII up to 127.

- UTF-8 the dominant encoding on the World Wide Web (*used in over 92% of websites*), uses one byte for the first 128 code points, and up to 4 bytes for other characters. The first 128 Unicode code points are the ASCII characters, which means that any ASCII text is also a UTF-8 text.
- UTF-16 uses 16bits to represent each character. This means that it is capable of representing 65,536 different characters.
- UTF-32 used 32bits to represent each character, meaning it can represent a character set 4,294, 967,298 possible characters, enough for all known languages.

character	encoding	bits
A	UTF-8	`01000001`
A	UTF-16	`00000000 01000001`
A	UTF-32	`00000000 00000000 00000000 01000001`
あ	UTF-8	`11100011 10000001 10000010`
あ	UTF-16	`00110000 01000010`
あ	UTF-32	`00000000 00000000 00110000 01000010`